ε-TEX and $\mathcal{NTS}$: A Status Report

Philip Taylor,
Technical Director, $\mathcal{NTS}$ project
The Computer Centre, RHBNC
University of London, U.K.
P.Taylor@Vax.Rhbnc.Ac.Uk

Abstract

The $\mathcal{NTS}$ project was created under the ægis of DANTE during a meeting held at Hamburg in 1992; its brief was to investigate the possibility of perpetuating all that is best in TEX whilst being free from the constraints which TEX's author, Prof. Knuth, has placed on its evolution. The group is now investigating both conservative and radical evolutionary paths for TEX-derived typesetting systems, these being respectively ε-TEX (extended TEX) and $\mathcal{NTS}$ (a New Typesetting System). The group is also concerned that whilst TEX itself is completely stable and uniform across all platforms, the adjuncts which accompany it vary from implementation to implementation and from site to site, and has therefore proposed that a 'canonical TEX kit' be specified, which once adopted could safely be assumed to form a part of *every* TEX installation. Work is now well advanced on the ε-TEX project, whilst the group are concurrently involved in identifying the key components of a complete portable TEX system and in investigating sources of funding which will allow the $\mathcal{NTS}$ project to become a reality.

Background

The $\mathcal{NTS}$ project first saw the light of day at the Hamburg meeting of DANTE during 1992; prior to this meeting, Joachim Lammarsch had sent e-mail messages asking those interested in the future of TEX to register their interest with him, and invitations to attend the DANTE meeting were sent to those who had registered their interest. At the meeting, Joachim proposed the formation of a working group, under the technical direction of Rainer Schöpf and with membership drawn from DANTE and UK-TUG, whose brief would encompass investigating, and possibly implementing, a successor or successors to TEX: that is, typesetting systems which would embody all that was best in TEX whilst being free from the constraints which had been placed on the evolution of TEX itself. These constraints had been imposed when Knuth announced that his work on TEX was complete, and that he now desired it to remain unchanged, apart from any essential bug fixes, in perpetuity.

Because of Rainer's very heavy commitments on other projects (and in particular on the LATEX3 project), not a lot was accomplished during the first year, and at the meeting of DANTE one year later Rainer announced that he was standing down as technical director; he and Joachim had already asked if I would be prepared to undertake that rôle, to which I had agreed, and therefore with effect from the 1993 Chemnitz meeting of DANTE the original $\mathcal{NTS}$ group was stood down and a new group formed.

The Group Meets

The initial membership of the re-formed group was just Rainer, Joachim and myself, and the first task was to identify others who would be willing to devote quite considerable amounts of time to ensuring the success of the $\mathcal{NTS}$ project. In the end Peter Breitenlohner, Bogusław Jackowski, Mariusz Olko, Bernd Raichle, Marek Ryćko, Joachim Schrod, Friedhelm Sowa and Chris Thompson were identified as likely candidates and invited to an inaugural meeting to be held in Kaiserslautern. (Barbara Beeton was also invited to join the group, but indicated in her acceptance that she would prefer to be a 'corresponding member', participating by e-mail but not in person.) Not all of those invited could attend, and the inaugural meeting was eventually attended by Peter B., Mariusz O., Bernd R., Joachim L., Joachim S., Friedhelm S. and myself, with Marion Neubauer acting as minutes secretary. The meeting was opened by Joachim L., who explained how $\mathcal{NTS}$ had come into existence, and the discussion was then opened for suggestions from those present.

Joachim S. was the first to speak, and he presented some quite radical ideas which he had clearly spent considerable time in preparing. The essence of his proposal was that TEX *per se* was no longer capable of evolution: quite apart from the constraints placed on its evolution by DEK, the complexity of the code and the interdependencies which existed within it militated against any further significant development; indeed, it was Joachim's contention that even Don no longer found it easy to modify the source of TEX, and that if any real evolution was to be achieved, the first task would involve a complete re-implementation. Joachim proposed that this be accomplished in two stages: (1) a re-implementation in a modern rapid-prototyping system such as CLOS ('Common Lisp Object System'), and (2) a further re-implementation in a modern efficient mainstream language such as 'C++'. Both of these re-implementations would be tackled using literate programming techniques.

Joachim explained the rationale behind the two-phase approach: in the first phase, the primary objective would be to identify the true modular nature of the TEX typesetting system, and to factor out into separate modules each of the fundamental components. This would allow, in the future, any component of the typesetting system to be replaced by an experimental version, with minimal impact on the design and interaction of the other modules; through this mechanism, alternatives to the current TEX algorithms could be evaluated and tested. For the initial implementation, however, this flexibility would not be fully exploited; instead, the system would simply be a fundamental re-implementation of TEX using modern object-oriented techniques, within which the modular nature of TEX would be properly represented. Although not expected to be efficient, this re-implementation would provide full TEX functionality, and once working the next step would be to demonstrate the functional equivalence of the re-implementation and TEX; this would be accomplished by means of the standard 'Trip' test for TEX, together with as much additional testing as was felt necessary to demonstrate that it was indeed a true 'TEX'.[1]

Once a faithful re-implementation of TEX had been achieved using the prototyping system, the second phase would involve a further re-implementation, this time using a widely available language such as 'C++'. The modular structure identified during the prototyping phase would be accurately mirrored in the 'production' phase, and each module would be functionally equivalent in both systems. Again a rigorous programme of testing would be required to demonstrate that the production system was also a true 'TEX'; once this was accomplished, the production system would be made available to TEX implementors world-wide, with a request that they attempt a port to the operating system(s) which they supported. Obviously the group would need to be responsive to problem reports from the implementors, and the ported implementations themselves would need to be tested for complete compliance with the TEX standard, but once these steps were accomplished, the project would be ready to move on to the next phase, which would be to release the re-implementation in its production form to the TEX world.

If all of this could be achieved within a reasonable timescale (measured in months rather than years), it was hoped that the TEX world could be encouraged to standardise on the re-implemented production TEX rather than TEX *per se*; for this to be accomplished, the re-implemented system would need to be at least as efficient as present TEX implementations, and equally bug-free. But if these criteria could be met, and if 're-implemented TEX' achieved the universal acceptance which was hoped for, then the group could turn its attention to the next and most exciting phase, which would be to start work on $\mathcal{NTS}$ itself, using the prototyping system to evaluate alternative typesetting paradigms and algorithms, and subsequently re-implementing the most successful of these and incorporating them into the production system.

Needless to say, the group were impressed by Joachim's proposal: clearly thought out, and quite radical in its approach, it would require considerable resources to be brought to fruition, yet the end results would almost certainly justify the means. However, the practical aspects could not be ignored, and the group agreed that without adequate financial backing they lacked the resources necessary to accomplish even phase-1, let alone subsequent phases. A small group of competent programmers, working full time for two to four months, would probably be needed to accomplish the first re-implementation, and perhaps only slightly fewer resources would be needed to accomplish phase-2. Bearing this caveat in mind, it was decided to put Joachim's plan on ice while seeking funding which would allow its commencement, and to identify fall-back plans which could be accomplished within the present resources of the group.

[1] The group are aware, of course, that only the *presence* of bugs can be demonstrated by testing, never their absence...

Two basic ideas were proposed: (1) despite Joachim's warning that TEX in its present form was essentially incapable of significant further development, Peter Breitenlohner felt that his experience in implementing TEX-XET and TEX--XET would allow him to make further changes to TEX within the framework of its present implementation; a number of good ideas had been proposed on the NTS-L list for incremental improvements to TEX, and both Peter and Bernd had ideas of their own which they would like to see implemented. (2) Marek and Bogusław had proposed during a conversation at Aston that before the $\mathcal{NTS}$ project sought to extend TEX in any way, it would be a very good idea to ensure that TEX's present capabilities were capable of being exploited to the full; in particular, they felt that TEX was frequently under-exploited because the additional software which was sometimes necessary to fully exploit TEX was not universally available. They therefore proposed that the group specify a minimum TEX kit which should be available at *every* site; once this was known to be universally available, TEX documents could be written to assume the existence of this kit, rather than simply assuming at most TEX + LATEX + the 75 Computer Modern fonts, which assumption tended to form the basis for portability at the moment.

Both of these proposals were well received: it was agreed that Peter Breitenlohner should take primary responsibility for extending TEX in its present form, whilst Marek and Bogusław would be asked if they were prepared to oversee specification of 'the canonical TEX kit' (it will be remembered that M&B were not present at the meeting, and therefore no assumptions could be made about their involvement in the project). All members of the group would be invited to contribute to all three projects, and members were also asked to investigate possible locations where a small team of programmers could work on $\mathcal{NTS}$. Such a location would clearly need good Internet connectivity (particularly as it was envisaged that not all members of the $\mathcal{NTS}$ programming team would necessarily work in the same place), and for reasons of economy it was considered desirable to site them at locations where their day-to-day expenses would not be too great.

Finally it was agreed that the membership of the group could be usefully enhanced by inviting some eminent members of currently unrepresented groups to participate.

The Interregnum

There then followed a period of several months during which members of the group returned to their normal place(s) of work; communication between members of the group was conducted by e-mail, and a report of the inaugural meeting published to encourage outside participation. Invitations to join the $\mathcal{NTS}$ project were extended to the nominated national representatives (sadly, not all had the courtesy to reply), and Bernd and Peter attempted to focus discussion on the NTS-L list by proposing some concrete ideas for 'extended TEX' (which was by now referred to simply as e-TEX, following the nomenclature separately proposed by Frank Mittelbach and myself in our *TUGboat* papers on the future of TEX.

During this interval, Joachim Schrod tendered his resignation from the group; all were sad to see him depart, particularly since his proposals for $\mathcal{NTS}$ still formed a central element of the group's plans, but the group respected his decision and since then he has pursued his own independent research into literate programming, typesetting systems, and the many other fields which are of interest to him professionally. Jiří Zlatuška was nominated by the Czech TEX User Group ($\mathcal{C}_S$TUG) to represent them on $\mathcal{NTS}$, and Klaus Lagally and Richard Palais accepted an invitation to join, the latter electing to follow Barbara's example and be a 'corresponding member'.

The Second Meeting

In February of 1994, DANTE held a further meeting, this time at Münster in Westfälische, and for a third time the $\mathcal{NTS}$ group were invited to be DANTE's guests; on this occasion none of the Polish delegates were able to be present, but Jiří represented the Czechs and Volker Schaa very kindly stepped into the breach to act as minutes secretary, Marion being unable to attend. Others present at the meeting included Peter Breitenlohner, Joachim Lammarsch, Bernd Raichle, Rainer Schöpf, Friedhelm Sowa and myself.

The group had made some promising contacts concerning possible funding for the project, but nothing definite had been agreed and commercial confidentiality regrettably dictates that no further detail can be recorded here. Following a very useful contact made at SOFSEM '93 (the Czech and Slovak annual computer science conference), some discussion had taken place concerning the possible use of

a semi-automatic system for reverse-engineering TEX, but again nothing had been agreed at the time of the meeting. Marek and Bogusław had been unable to offer a committment to leading the 'canonical TEX project', and no further progress had been made in that area, despite (or perhaps because of) a counter-proposal by TUG who had somewhat belatedly decided to undertake an almost identical scheme.

By far the most promising news was that Peter Breitenlohner had made enormous progress in the implementation of ε-TEX, and indeed had a preliminary version already working. In view of this, and in view of the fact that the membership of the group had undergone some very significant changes, it was decided to adopt a slightly more formal structure within the group: Joachim Lammarsch would continue to take overall financial and political responsibility, whilst I would continue as Technical Director; Peter Breitenlohner would head the ε-TEX sub-group, backed up by Bernd Raichle, whilst Jiří Zlatuška would head the $\mathcal{NTS}$ sub-group, again backed up by Bernd; Rainer Schöpf would take responsibility for the 'canonical TEX kit' project, backed up by Friedhelm Sowa, and in addition Friedhelm would continue to act as treasurer. No formal responsibilities were laid on members who were not present at the meeting, although it was hoped that Mariusz Olko would continue to look after the multi-lingual areas of both ε-TEX and $\mathcal{NTS}$ by liaising with the TWG-MLC.

As progress on the 'canonical TEX kit' project had been almost non-existent, the group felt it worthwhile to devote some time to attempting to identify the elements of a TEX system which were truly fundamental. Rather interestingly, members seemed to hold stronger (and sometimes more divergent) views on this subject than on almost anything proposed for either ε-TEX or $\mathcal{NTS}$! The net result was that only the most basic elements were agreed, and considerable further work will be needed in this area in conjunction not only with TUG but also with the entire TEX community.

Much of the discussion which took place during the $\mathcal{NTS}$ meeting at Münster concerned specific details of proposals for ε-TEX, and whilst these formed the basis for progress, later discussions (both in less formal meetings at Münster and subsequently via e-mail) caused considerable revision of the ideas which emerged; what follows is therefore a synthesis of ideas which were first mooted at the $\mathcal{NTS}$ meeting in Münster, together with ideas which were mooted later, either in less formal meetings or via e-mail.

ε-TEX: Some Specific Proposals

Perhaps the most important of the proposed extensions is the mechanism by which the extensions themselves are activated, either individually or as a group; an absolutely fundamental requirement is that ε-TEX be capable of processing all existing TEX documents in a manner identical to TEX. The typeset results must be identical to those produced by TEX, as must all 'reasonable' side effects (for example, information written to ancillary files). Thus it is intended that there be no possible reason for the non-adoption of ε-TEX as a replacement for TEX. The mechanism by which divergent behaviour is enabled will be under user control: a user may elect to use ε-TEX in a totally compatible manner, or may elect to use only that set of extensions which do not compromise the semantics of TEX, or may elect to use one or more of the most radical extensions (e.g. TEX--XET, the left-right/right-left extension for non-European languages) which by their very nature require a fundamental modification to the behaviour of the typesetting system.

Once such a mechanism is in place, users (or more precisely user documents) will need to be able to investigate their environment; since user-X may habitually use extension-A, yet may send his/her document in source form to user-Y who habitually disables extension-A, a document must be able to check which extensions are available, and to either adopt a fall-back position if a preferred extension is not available, or to issue an error message and abort tidily (a user document may, of course, attempt to enable an extension which it needs, but the result may be 'not available within this environment'). A mechanism for environmental enquiries is therefore proposed.

Other proposals for ε-TEX include: (a) improved control over tracing (TEX can be very verbose, and interpreting the trace output is distinctly non-trivial); (b) an additional class of maths delimiters (middle, as well as left and right); (c) improved access to the current interaction mode (to allow code to ascertain as well as change the current mode of interaction); (d) improved mechanisms for checking the existence of a control sequence without necessarily creating such a sequence; (e) improved avoidance of internal integer overflow; (f) an alternative ligature/kerning implementation; (g) extensions to the set of valid prefixes for macro definitions, such as `\protect` (to inhibit subsequent undesired expansion) and `\bind` (to render a definition independent of the environment within which it is expanded); (h) support for colour.

$\mathcal{NTS}$

The complete re-implementation of TEX ('NTS') is a far more ambitious project, the success of which is crucially dependent on obtaining adequate funding. The proposals which follow should therefore be regarded as being preliminary ideas, rather than absolute decisions which have been cast in stone. Any ideas which the TEX community in general would like to contribute to the project will be very gratefully received!

In phase-1, a rapid prototyping language such as CLOS ('Common Lisp Object System') or PROLOG will be used to develop a primary re-implementation, within which it will be possible to experiment with various possible internal modular representations of the present TEX typesetting engine; these alternative representations will be evaluated to attempt to determine an 'ideal' model of the TEX engine, within which the various functional elements are as independent as possible. The purpose of this analysis is to allow each of these functional elements to be enhanced or replaced at will, with minimal effect on the other modules; such flexibility is demonstrably lacking in the present TEX implementation, and is the primary reason for proposing a complete re-implementation. Once an ideal representation has been found, the work will progress to the second phase, although the work invested in developing the first phase will not be wasted: there are many reasons why the phase-1 implementation will continue to be both needed and useful.

Firstly, prototyping languages based on term data structures, used in conjunction with a disciplined programming style free from the exploitation of side-effects, allow the generation of code which would provide a specification of TEX independent of the procedural Pascal source code, and potentially amenable to the use of automated transformation techniques; such code could also be partially interpreted or meta-interpreted. This would produce an environment within which the development of enhancements is much facilitated, reaching beyond the particular structure of the code of the program.

Secondly, for some of these prototyping languages at least, rather powerful enhancements have been developed, in particular the employment of constraint solving in constraint logic programming languages. This could allow certain parts of the typesetting task to be formulated as a set of constraints to be fulfilled, within which constraints an optimal solution must be found. Certain classes of problem, particularly those occurring in connection with chapter layout optimisation, float/insertion placement, and conformance to the grid, all appear to require a search of the space state and the employment of backtracking within certain parts of the typeset text.

Thirdly, the kind of test-bed implementation proposed for phase-1 should provide a suitable basis for the independent exploration of possible enhanced designs and/or implementations, including alternative attempts to provide solutions based on different approaches and paradigms, and also including investigations into the trade-offs involved when selecting from a set of mutually incompatible features.

In phase-2, the modular structure which has been identified in phase-1 will be once again re-implemented, this time in a widely available compiled language such as C++. This re-implementation is aimed entirely at efficiency: the test-bed which is developed during phase-1 is fully expected to be unacceptably inefficient as a production typesetting system, and the flexibility which will characterise it is unlikely to be of any use in a production environment. Thus in phase-2 attention will be paid to the efficiency of the algorithms used, and particular attention will be paid to ensuring that the resulting system is able to run efficiently on a very wide range of hardware and software platforms. Whilst ε-TEX is expected to run on the same range of hardware and software as the present TEX system (i.e., everything from an AT-class PC to a Cyber or Cray mainframe), it is accepted that NTS may have slightly reduced availability: none the less, the minimum configuration on which NTS will be expected to run efficiently is an 80386-class PC (or the equivalent in other architectures); everything above should pose no problems.

Both the phase-1 and phase-2 implementations, in their initial release, will be purely and simply re-implementations of TEX (or, more likely, of ε-TEX, since it is hoped that by the time phase-1 is complete, ε-TEX will have gained widespread adoption amongst the TEX community; the intellectual effort invested in extending and enhancing TEX to create ε-TEX will therefore not be wasted). These re-implementations will be rigorously tested to ensure that they behave identically to (ε-)TEX in all circumstances, and only once this testing is complete will they be offered to the community at large.

Once the project has reached this stage, and the phase-2 re-implementation has been made available to, and used by, a significant proportion of the existing TEX user community (and any other interested parties), the real research work will commence: using the phase-1 re-implementation (the 'test bed', as

it is generally termed), work will commence on investigating alternatives to the existing TEX paradigms. For example, one of the frequent criticisms levelled against TEX is that its command line oriented user interface seems unbelievably anachronistic to those whose experience of computers is almost entirely limited to graphical user interfaces (GUIs); whilst various implementors (and most notably Blue Sky Research, with their 'Lightning TEXtures' implementation) have endeavoured to conceal this interface beneath a more graphical front-end, all of these attempts have been restricted to one particular hardware and software platform. A primary longer-term objective of the $\mathcal{NTS}$ project would therefore be to provide an integral graphical interface to $\mathcal{NTS}$ in an entirely portable manner; this interface would not be layered on top of the existing command line interface, but would operate at the same hierarchical level, thereby allowing the more GUI-oriented to use $\mathcal{NTS}$ in their preferred manner with no loss of efficiency (the existing command line interface would not necessarily disappear: the group are well aware that one reason for the widespread adoption of TEX may well lie in the appeal of its more traditional interface to large numbers of people who pre-date the GUI era, as well as in the re-creatability and reproducibility of documents which are represented as human-readable text rather than as ephemeral mouse movements on a screen).

There are many other areas in which TEX is felt to be deficient by a significant group of well-informed users; various papers have been published in *TUGboat* in which some of these deficiencies have been discussed. It is therefore proposed that NTS would attempt to rectify as many of these deficiencies as possible by providing some or all of the following features: (a) the ability to typeset material on a grid; (b) the ability to flow text around regular (and irregular) insertions; (c) the treatment of 'the spread' (two facing pages) as the basic unit of makeup; (d) the treatment of the chapter (in book-like material) as the minimum unit over which page optimisation should be performed; (e) provision of pattern recognition within the paragraph-building algorithm, which could enable both the avoidance of 'rivers' (accidental contiguous regions of regular white space spanning several lines) and also (at a less graphical level) the avoidance of subtly different hyphenations on consecutive lines; (f) improved interaction between the line- and page-breaking algorithms, to permit co-optimisation; (g) an improved model of lines within a paragraph, to enable languages which make regular use of diacritical marks to be set on a tighter leading (interline spacing) without increasing the risk of conflict between superior and inferior diacritics; (h) intrinsic support for hanging punctuation (whereby leading and trailing punctuation on a line are allowed to hang out into the margin); (i) improved interaction between the page makeup module and the paragraph building module, to allow the shape of a paragraph to be influenced by its final position on the page; (j) greater awareness within the typesetting engine of the shape of a glyph (character), which could allow spacing to be better optimised; (k) improved parameterisation of fonts; (l) improved access to the ligature and kerning information from within the typesetting system. There are many other areas, primarily concerned with rather technical aspects of TEX-the-language and TEX-the-typesetting-system, which have also been convincingly argued are less than perfect, and which it is intended will be addressed by $\mathcal{NTS}$.

The long-term objective of $\mathcal{NTS}$ is therefore to make maximum use of the test bed to investigate and evaluate possible approaches to overcoming the various perceived deficiencies of TEX, and to incrementally produce an ever-better typesetting system, capable of taking maximum advantage of current technology. This typesetting system will undoubtedly become ever less TEX-like, yet the group believe that there are so many good ideas enshrined in TEX that the day when $\mathcal{NTS}$ owes nothing to TEX lies several decades in the future.

The author would like to express his sincere thanks to Peter Breitenlohner and Jiří Zlatuška for their most helpful comments on the first draft of this article, and to Michel Goossens for his willingness to allow late and unrefereed submission of the copy; any errors remaining are, of course, entirely the responsibility of the author.